

# Concepte și aplicații în Vederea Artificială

## Laborator 1

### Obiectiv:

În acest laborator vom învăța să lucrăm cu bibliotecă OpenCV din Python.

#### 1.1 Instalarea bibliotecii OpenCV

Deschideți *Anaconda prompt* și tastați:  
**pip install opencv-python**

#### 1.2 Importarea bibliotecii

```
import cv2 as cv  
print('versiune: ', cv.__version__)
```

#### 1.3 Citirea unei imagini color

```
image = cv.imread('image_name.png')
```

Variabila *image* este de tipul *numpy-array uint8*. Dimensiunea variabilei este de: H, W, 3, unde H este înălțimea imaginii, W este lățimea, iar 3 este numărul de canale RGB.

#### 1.4 Citirea unei imagini în tonuri de gri

```
image = cv.imread('image_name.png', cv.IMREAD_GRAYSCALE)
```

Dimensiunea variabilei este de: H, W.

#### 1.5 Afișarea unei imagini

```
cv.imshow('windowName', image)  
cv.waitKey(0) % se așteaptă apăsarea unei taste  
cv.destroyAllWindows() % se închide fereastra
```

#### 1.6 Operații cu o matrice de intensități

Fie *img* un *numpy-array*  $100 \times 100$  de tipul *uint8* obținută astfel:

```
img = cv.resize(cv.cvtColor(cv.imread('football.jpg'), cv.COLOR_BGR2GRAY), (100, 100));
```

ce corespunde unei imagini de intensităţi. Scrieţi codul Python ce rezolvă următoarele cerinţe:

- (a) Sortaţi elementele/intensităţile din matricea *img* (folosiţi funcţia *np.sort* din librăria Numpy), punând elementele sortate în vectorul *x* de dimensiuni  $10000 \times 1$  (10000 de linii, o coloană). Plotaţi apoi valorile din *x* (ca funcţie de indecşii lui *x*, folosind librăria Matplotlib).
- (b) Afişaţi submatricea din *img* de dimensiuni  $50 \times 50$  care corespunde sfertului matricei *img* din partea dreapta-jos.
- (c) Aflaţi pragul de intensitate *t* cu proprietatea următoare: jumătate din elementele matricei *img* au o valoare  $\leq t$  (puteţi folosi funcţia *np.median*).
- (d) Creaţi şi afişaţi matricea/ imaginea *B*, de aceleaşi dimensiuni ca matricea *img* obţinută din *img* astfel: fiecare pixel din *B* are culoarea albă (intensitate = 255) dacă pixelul corespunzător din *img* are intensitatea  $\geq t$ , altfel are culoarea neagră (intensitate = 0).
- (e) Creaţi şi afişaţi matricea/ imaginea *C*, de aceleaşi dimensiuni ca matricea *img* obţinută din *img* astfel: fiecare pixel din *C* are intensitatea egală cu pixelul corespunzător din *img* din care se scade intensitatea medie a imaginii *img*; pixelii astfel obţinuţi cu intensitatea  $< 0$  vor fi setaţi ca având intensitatea = 0.
- (f) Aflaţi valoarea minimă (*np.min*) a intensităţi şi afişaţi poziţiile în care apare (*np.where*) în *img*.

### 1.7 Imaginea medie a unei colecţii de imagini

O colecţie de imagini similare poate fi descrisă prin imaginea ei medie. Figura 1 ilustrează patru astfel de imagini medii, foarte sugestive pentru descrierea fiecărei colecţii în parte. Scopul acestui exerciţiu este de a scrie o funcţie Python ce realizează o operaţie similară.



Figura 1: Exemple de imagini medii ale unor colecţii de imagini similare.

Downloadaţi arhiva *colectiilmagini.zip* ce însoţeşte acest laborator. Arhiva conţine două colecţii de imagini grupate în două directoare: *set1* şi *set2*. Toate imaginile din fiecare

colecție au aceleași dimensiuni și conțin un obiect centrat al aceleași clase de obiecte (set1 - vapor, set2 - avion).

Scrieți o funcție Python care, pentru o colecție de imagini dată, realizează următoarele:

- a) calculează imaginea medie color a colecției (`img_color`);
- b) calculează imaginea medie de intensitate a colecției (`img_gray`);
- c) calculează matricea  $X$ , fiecare element  $X[i,j]$  al matricei reprezintă deviația standard a intensităților pixelilor  $(i,j)$  din imaginile de intensitate corespunzătoare imaginilor colecției;
- d) afișează cele trei imagini (`img_color`, `img_gray`,  $X$ ) calculate.

Funcția ar trebui să arate astfel:

```
def colectie_imagini(dir_path);
```

unde *dir\_path* reprezintă directorul unde se afla colecția de imagini.

Aplicați funcția scrisă pentru a obține imaginile pentru cele două colecții. Afișați imaginile și comentați felul în care arată.

Funcții folosite: *np.mean*, *np.std*, *cv.cvtColor*, *os.listdir*, *cv.imshow*.

### 1.8 Modificarea unei imagini

Citiți imaginea *butterfly.jpeg* în variabila *img*. Extrageți 500 de sub-imagini de dimensiune  $20 \times 20$  din poziții aleatorii ale imaginii *img*. Modificați porțiunea din imagine care are colțul stang-sus în poziția  $(250, 250)$  cu cea mai apropiată sub-imagine (din cele 500) conform distanței  $L_2$ . Salvați imaginea obținută.



Figura 2: În partea stângă avem imaginea originală, iar în dreaptă este imaginea modificată.